Amendments to the Claims:

This listing of claims will replace all prior versions, and listings of claims in the application:

Listing of Claims:

1 - 34. (canceled)

35. (new) A binary translation system, comprising:

a non-optimizing foreign code execution module configured to maintain dedicated foreign state for each foreign binary operation executed allowing for the exceptions arisen to be handled precisely; and

an optimizing binary translator configured to translate foreign binary operations into optimized sequences of host operations in such a way as to improve the speed of execution of the sequences; and

a host CPU configured to execute the host operations; and

a documentation generator configured to generate documentation for optimized sequences of host operations, wherein the documentation describes operations required to calculate a corresponding foreign state for the appointed points;

a documentation tracker configured to track an executable path of optimized sequences of host operations and record host operation addresses at appointed points of the host operation sequences being executed; and

a recovery mechanism configured to use the documentation to continue foreign codes execution based on information in the documentation in case of the exception arisen during the execution of the corresponding optimized host codes.

36. (new) The binary translation system of claim 35, wherein the sequence of host operations consumes input values from the dedicated foreign state and puts output results into the said foreign state.

37. (new) The binary translation system of claim 36, wherein the dedicated foreign state is not changed in case of exception arisen during execution of the host operations sequence.

38. (new) The binary translation system of claim 37, wherein a foreign exception handler is executed in response to the exception occurring during execution of the host operations sequence, the foreign exception handler configured to take input values from the unchanged dedicated foreign state.

39. (new) The binary translation system of claim 35, wherein the optimizing translator is configured to combine all host operations sequences corresponding to a number of foreign binary operations into one executable unit.

40. (new) The binary translation system of claim 39, wherein the optimizing translator removes intermediate outputs from the dedicated foreign state for each sequence of host operations and saves information about the intermediate outputs in the documentation.

41. (new) The binary translation system of claim 40, wherein the documentation describes a virtual foreign state between each sequence of host operations.

42. (new) The binary translation system of claim 41, wherein the optimizing translator preserves values from each virtual state in temporary registers without re-using them until next virtual foreign state is passed.

43. (new) The binary translation system of claim 42, wherein every memory write operation changes unconditionally the dedicated foreign state and is considered as a possible recovery point in case of exception.

44. (new) The binary translation system of claim 43, wherein the recovery point corresponds to one of the virtual foreign states as described in the documentation.

45. (new) The binary translation system of claim 44, wherein the documentation is a host to foreign registers computation rules table or host binary code used to recompute the foreign state.

46. (new) The binary translation system of claim 35, wherein the host CPU is configured to execute host operations and produces exceptions as if it were a foreign CPU.

47. (new) The binary translation system of claim 35, wherein documentation tracker is a special host operation, a recovery point register and an apparatus for saving instruction address in the recovery point register in response to executing the recovery point operation.

48. (new) The binary translation system of claim 35, wherein the recovery mechanism is a host program configured to recompute a correct dedicated foreign state according to the documentation in the case of an exception generated.

49. (new) A method of recomputing a dedicated foreign state in a binary translation system from documentation generated by an optimizing translator in a case of an exception arising during execution of optimized binary translated code translated from a foreign code, the method comprising:

designating a set of recovery points in the optimized binary translated code during optimized translation of the foreign code, wherein each recovery point represents a foreign state;

generating a set of documentations during the optimized translation of the foreign code, wherein each documentation in the set of documentations corresponds to a recovery point in the optimized binary translated code and describes operations required to calculate a corresponding foreign state for the recovery point; and using one of the documentations in the set of documentations corresponding executed optimized binary translated code when an exception arises during its execution to recover a foreign state corresponding to a recovery point for the exception.

50. (new) The method of claim 49, wherein the documentation describes a virtual foreign state and is discovered by using a recovery point register.

51. (new) The method of claim 50, wherein the optimizing translator sets recovery point operations in the host code in such a way that there is no unconditional changes in the dedicated foreign state between adjacent recovery points.

52. (new) The method of claim 51, further comprising:

detecting an exception in the optimized binary translated code;

reading the recovery point register;

finding the appropriate documentation in the set of documentations corresponding to contents of the recovery point register;

recomputing a foreign state according to the documentation;

switching execution of the foreign code to non-optimizing execution;

executing each foreign code operation in sequence;

updating a dedicated foreign state at the end of each foreign code operation;

detecting an exception during the course of non-optimizing execution; and

transferring contents of the dedicated foreign state to the foreign exception handler.

53. (new) An apparatus for a host microprocessor, the apparatus comprising:

a recovery point register; and

means for saving a host instruction address into the recovery point register.

54. (new) The apparatus of claim 53, wherein the recovery register is a dedicated host register.

55. (new) The apparatus of claim 53, wherein the means for saving a host instruction address into the recovery point register comprises a dedicated host microprocessor instruction.

56. (new) The apparatus of claim 53, wherein the host instruction is stored in the host microprocessor memory together with a set of tag valuesthat are associated with the host microprocessor instruction, wherein the means for saving a host instruction address into the recovery point register comprises a specific tag value associated with the host microprocessor instruction being executed.